Int J Parallel Prog (2017) 45:1026–1045
DOI 10.1007/s10766-016-0453-2

Energy-Aware Modeling of Scaled Heterogeneous Systems

Ami Marowka[1]

Received: 30 January 2016 / Accepted: 22 September 2016 / Published online: 27 September 2016

Abstract Many-core processors are accelerating the performance of contemporary high-performance systems. Managing power consumption within these systems demands low-power architectures to increase power savings. One of the promising solutions offered today by microprocessor architects is asymmetric microprocessors that integrate different core architectures on a single die. This paper presents analytical models based on *scaled* power metrics to analyze the impact of various architectural design choices on *scaled* performance and power savings. The power consumption implications of different processing schemes and various chip configurations were also analyzed. Analysis shows that by choosing the optimal chip configuration, energy efficiency and energy savings can be increased considerably.

Keywords Energy efficiency · Gustafson–Barsis's law · Hybrid architecture · Performance per Watt · Modeling techniques

1 Introduction

The major challenge that microprocessor designers will face in the coming decade is not just power, but also energy efficiency. Upcoming new mobile devices will consume more power, while supercomputers in the foreseeable future will consume hundreds of megawatts of power. Although Moore's Law [1] continues to offer solutions with more transistors, power budgets limit our ability to use them.

However, there are promising solutions such as heterogeneous many-core architectures that will provide higher performance at lower energy requirements and

✉ Ami Marowka
amimar2@yahoo.com

[1] Parallel Research Lab, Tel Aviv, Israel

reduced leakage. Recent research shows that integrated CPU–GPU processors have the potential to deliver more energy efficient computations, which is encouraging chip manufacturers to reconsider the benefits of heterogeneous parallel computing. The integration of CPU and DSP cores on a single chip has provided an attractive solution for the mobile and embedded market segments, and a similar direction for CPU–GPU computing appears to be an obvious move. It is known that the integration of thin cores and fat cores on a single processor achieves a better performance gain per watt. For example, a study of analytical models of various heterogeneous multi-core processor configurations found that the integration of many simplified cores in a single complex core achieved greater speedup and energy efficiency when compared with homogeneous simplified cores [2]. Thus, it is generally agreed that a heterogeneous chip integrating different core architectures, such as CPU and GPU, on a single die is the most promising technology [3–8]. Chip manufacturers such as Intel, NIVIDIA, and AMD have already announced such architectures, i.e., Intel Sandy Bridge, AMD's Fusion APUs, and NVIDIA's Project Denver.

Intel researchers have shown that integration of general-purpose cores alongside special-purpose hardware accelerators can improve energy efficiency by an order of magnitude [9]. Based on their view, the future tera-scale processor will contain a few dedicated hardware accelerators such as speech recognition accelerators, GPU accelerators, and encryption accelerators that operate at ultra-low voltage (down to 320 mV) and ultra-low frequency (down to 23 MHz) and consume ultra-low power (down to 56 μW).

Despite some criticisms [10,11], Amdahl's law [12] and Gustafson–Barsis's Law [13,14] are still relevant at the dawn of a heterogeneous many-core computing era. Both laws are simple analytical models that help developers to evaluate the actual speedup that can be achieved using a parallel program. They represent two points of view that are not contradictory, but rather complement each other. However, neither of these laws is perfect. Amdahl's Law and Gustafson–Barsis's Law do not account for overheads associated with the creation/destruction of processes/threads and with maintaining cache coherence. Neither do they account for other types of serial tasks such as identification of critical sections, synchronization, lock management, and load balancing. Taking in account all these issues for making performance analysis is very difficult. *Therefore, to keep our analytic models simple, as Amdahl's Law and Gustafson–Barsis's Law, the extensions proposed in this paper do not incorporate the parallel computations and communication overheads. Our models incorporate only a few factors* (Table 1) *that are necessary for energy efficiency analysis of dual-architecture systems.*

Amdahl's Law is based on the assumption that a problem of fixed size is being solved in parallel, while Gustafson–Barsis's Law is based on the assumption that a problem of fixed time is being solved in parallel. Therefore, depending on the application domain, either Amdahl's or Gustafson–Barsis's assumptions might be valid. For example, extremely parallel computations such as in image rendering are processing many pixels simultaneously and independently. As the number of cores increases, so does the problem size, and the inherently serial portion becomes much smaller as a proportion of the overall problem. Because Amdahl's Law cannot address this relationship, Gustafson modified Amdahl's work to state that the overall problem size

Table 1 List of parameters appear in the formulas

c	The number CPU of cores
g	The number GPU of cores
f	The fraction of a program's execution time that is parallelizable
α	The fraction of a program's parallel execution time where the program runs in parallel on the CPU cores
β	A GPU core's performance normalized to that of a CPU-core
k_c	The fraction of power a single CPU core consumes in its idle state
k_g	The fraction of power a single GPU core consumes in its idle state
w_g	The active GPU core's power consumption relative to that of an active CPU-core

should increase proportionally to the number of cores, while the size of the serial portion of the problem should remain constant as the problem size increases.

Furthermore, the future relevance of the laws requires their extension by the inclusion of constraints and architectural trends demanded by modern multiprocessor chips. In [15,16], we extended Amdahl's Law according to the work of Woo and Lee [2] and applied it to the case of a hybrid CPU–GPU multi-core processor. In this work we extend Gustafson–Barsis's Law by using the same methodology as we done in [15–17], and applied it to the case of a hybrid many-core processor. Since we are using the same methodology, part of the wording of the issues is the same but the formulas and the equations are different. In Sect. 2 we elaborate about our previous work in [15,16] and in Sect. 7 we compare the results of [15,16] against the results of this study.

Core contributions of this paper are as follows:

- To define and formulate three metrics: scaled speedup, scaled performance per watt, and scaled performance per joule.
- Using the above metrics, to evaluate the energy efficiency and scalability of three processing schemes available for heterogeneous computing: symmetric, asymmetric and simultaneous asymmetric.
- For each processing scheme, to examine how performance, power and energy are affected by different chip configurations.
- Finally, to analyze and compare the outcomes of the three analytical models and to show how considerable energy savings can be achieved by choosing the optimal chip configuration.

The remainder of this paper is organized as follows. Section 2 presents an overview of the related work in this area. Section 3 presents an analytical model of a symmetric multi-core processor that reformulates Gustafson–Barsis's Law to capture power constraints. In Sect. 4 we continue by applying energy constraints to an analytical model of an asymmetric processor. In Sect. 5 we study how performance and power consumption are affected by chip configurations in simultaneous asymmetric processing. In Sect. 6 we compare the three analytical models. Section 7 is a discussion about future work and the findings found in this study compared to our previous work related to Amdahl's Law. Section 8 concludes the paper.

2 Related Work

In our previous work [15] we extend a study conducted by Woo and Lee [2] and apply it to the case of hybrid CPU–GPU multi-core processors. In [2] the authors study three chip configurations. (a) A symmetric many-fat-core processor that replicates a superscalar processor on a single chip. (b) A symmetric many-thin-core processor that replicates a simplified power-efficient core on a single chip, and (c) an asymmetric many-thin core processor with simplified efficient cores and only one fat core as the host processor. *In our study we investigate chip configurations of different number of fat cores and thin cores on a single die.*

We investigate how energy efficiency and scalability are affected by the power constraints imposed on modern CPU–GPU based heterogeneous processors. We study how the performance per watt is affected by different CPU–GPU performance ratios, different values of active GPU cores power consumption relative to that of an active CPU-core, and different fractions of power a single CPU core and a single GPU-core consume in their idle states. *This part of the study does not appear in* [2].

We present analytical models that extend Amdahl's Law by accounting for energy limitations and we analyze three processing modes available for heterogeneous computing, i.e., symmetric, asymmetric, and simultaneous asymmetric. *In* [2], *the authors studied only the symmetric mode and the asymmetric mode for a single fat core.* We study the asymmetric mode for any number of thin cores and fat cores and present a new processing mode called *simultaneous asymmetric* where the CPU cores and the GPU-cores are executed simultaneously.

The simulations show that greater parallelism opportunities yield better speedup and offer more chip configuration choices, while encouraging the search for better scalable software with power saving. Simultaneous processing yields excellent speedup with a peak performance at a chip configuration with a single CPU-core. In contrast, asymmetric speedup delivers poor speedups at the extreme points where the number of CPU-cores is small or large, indicating that dynamic configuration is required to identify and set the optimal chip organization. The performance per watt in the three cases analyzed, i.e., symmetric(s), asymmetric (a), and simultaneous asymmetric (sa), show that simultaneous processing outperforms the other cases, with a peak performance at a chip configuration with a single CPU-core. This configuration also yields the best performance in the symmetric case, but it achieves 28 % less performance compared to the simultaneous asymmetric case. The results of the performance per joule show that greater parallelism yields better energy efficiency and offers more chip configurations choices, while encouraging the search or better scalable software with energy saving. Simultaneous processing yields an excellent performance per joule with peak performance using a chip configuration of a single CPU-core. In contrast, the asymmetric processor delivers poor performance per joule at extreme points where the number of CPU-cores is small or large, which requires that the dynamic configuration is identified and set for optimal chip organization.

Hill and Marty [18] studied the implications of Amdahl's law on multi-core hardware resources and proposed the design of future chips based on the overall chip performance rather than core efficiencies. The major assumption in that model was that a chip is composed of many basic cores and their resources can be combined

dynamically to create a more powerful core with higher sequential performance. Using Amdahl's law, they showed that asymmetric multi-core chips designed with one fat core and many thin cores exhibited better performance than symmetric multi-core chip designs. For example, with $f = 0.975$ (the fraction of computation that can be parallelized) and $n = 256$ (Base Core Equivalents), the best asymmetric speedup was 125.0, whereas the best symmetric speedup was 51.2. Individual core resources could be dynamically combined to increase performance of the sequential component, so the performance was always improved. In our example, the speedup was increased to 186.0.

Woo and Lee [2] developed a many-core performance per energy analytical model that revisited Amdahl's Law. Using their model the authors investigated the energy efficiency of three architecture configurations. The first architecture studied contained multi-superscalar cores, the second architecture contained many simplified and energy efficient cores, and the third architecture was an asymmetric configuration of one superscalar core and many simplified energy efficient cores. The evaluation results showed that under restricted power budget conditions the asymmetric configuration usually exhibited better performance per watt. The energy consumption was reduced linearly as the performance was improved with parallelization scales. Furthermore, improving the parallelization efficiency by load balancing among processors increased the efficiency of power consumption and increased the battery life.

Sun and Chen [19] studied the scalability of multi-core processors and reached more optimistic conclusions compared with the analysis conducted by Hill and Marty [18]. The authors suggested that the fixed-size assumption of Amdahl's law was unrealistic and that the fixed-time and memory-bounded models might better reflect real world applications. They presented extensions of these models for multi-core architectures and showed that there was no upper bound on the scalability of multi-core architectures. However, the authors suggested that the major problem limiting multi-core scalability is the memory data access delay and they called for more research to resolve this memory-wall problem.

Esmaeilzadeh et al. [20] performed a systematic and comprehensive study to estimate the performance gains from the next five multi-core generations. Accurate predictions require the integration of as many factors as possible. Thus, the study included: power, frequency and area limits; device, core and multi-core scaling; chip organization; chip topologies (symmetric, asymmetric, dynamic, and fused); and benchmark profiles. They constructed models based on pessimistic and optimistic forecasts, and observations of previous works with data from 150 processors. The conclusions were not encouraging. Over five technology generations only a 7.9x average speedup was predicted with multi-core processors, while over 50 % of the chip resources will be turned off due to power limitations. Neither multi-core CPUs nor many-core GPUs architectures were considered to have the potential for delivering the required performance speedup levels.

Cho and Melhem [21,22] studied the mutual effects of parallelization, program performance, and energy consumption. Their analytic model was applied to a machine that could turn off individual cores, while others do not make this assumption. The main prediction was that greater parallelism (a greater ratio of the parallel portion in the program) and more cores helped reduce energy use. Moreover, it was shown that

is possible to reduce the processor speeds and gain further dynamic energy reductions before static energy becomes the dominant factor determining the total amount of energy used.

Hong and Kim [23] developed an integrated power and performance modeling system (IPP) for the GPU architecture. IPP is an empirical power model that aims to predict performance-per-watt and the optimal number of active cores for bandwidth-limited applications. IPP uses predicted execution times to predict power consumption. In order to predict the execution time the authors used a special-purpose GPU analytical timing model. Moreover, to obtain the power model parameters, they designed a set of synthetic micro-benchmarks that stress different architectural components in the GPU.

The evaluation of the proposed model was done by using NVIDIA GTX280 GPU. The authors show that by predicting the optimal number of active cores, they can save up to 22.09 % of runtime GPU energy consumption and on average 10.99 % of that for five memory bandwidth-limited benchmarks. They also calculated the power savings if a per-core power gating mechanism is employed, and the result shows an average of 25.85 % in energy reduction. IPP predicts the power consumption and the execution time with an average of 8.94 % error for the evaluated benchmarks GPGPU kernels. It can be used by a thread scheduler in order to manage the power system more efficiently or by the programmers to optimize program configurations.

Pei et al. [24] present an enhanced performance-energy efficiency analytical model for integrated heterogeneous parallel multi-core system which takes into account the overhead cost of data preparation (i.e., accessing memory, communication on-chips or off-chips and synchronization among cores). Their analysis shows that higher parallelism gained from either computation or data preparation brings greater energy-efficiency.

Karanikolaou et al. [25] proposed an analytic modeling for evaluation the energy consumption of distributed and many-core platforms. They measured the power consumption of the processors in idle and fully utilized modes and compared the theoretical estimations to the experimental results using performance/power and performance/energy ratio metrics.

Kim et al. [26] studied the energy efficiency of the sequential part acceleration, and how to find out the optimal frequency boosting ratio which maximizes energy efficiency. The results show that energy efficiency of the acceleration increases with the number of cores and an optimal frequency boosting ratio can be determined.

3 Symmetric Processors

In this section we reformulate Gustafson–Barsis's Law to capture the necessary changes imposed by power constraints. We start with the traditional definition of a symmetric multi-core processor and continue by applying energy constraints to the equations following the method of Woo and Lee [2] and Marowka [15].

3.1 Symmetric Scaled Speedup

Gustafson–Barsis's Law begins with a parallel computation and estimates how much faster the parallel computation is than the same computation executing on a single core. Gustafson argues that, as processor power increases, the size of the problem set also tends to increase. This is why the speedup determined by Gustafson–Barsis's Law, also called *scaled speedup*, is the time required by a parallel computation divided into the time hypothetically required to solve the same problem on a single core.

According to the Gustafson–Barsis's Law, a typical program has a serial portion that cannot be parallelized (and therefore can be executed only by a single core) and a parallel portion that can be parallelized (and therefore can be executed by any number of cores in the processor). Let the parallel execution time of the program be normalized to 1, and let the serial and parallel portions be denoted by s and p respectively. Then the following equation concisely describes the law:

$$Scaled\ Speedup_s = s + (1 - s) \cdot c = c + (1 - c) \cdot s \tag{1}$$

where c is the number of cores and s is the fraction of a program's execution time that is spend in serial code ($0 \leq s \leq 1$).

Figure 1 shows the symmetric scaled speedup as a function of number of CPU cores for various values of s. Clearly, these graphs show that the performance result continues to scale upward as more processor cores are applied to the computational load.

3.2 Symmetric Scaled Performance Per Watt

To model power consumption in realistic scenarios, we introduce the variable k_c to represent the fraction of power a single CPU core consumes in its idle state ($0 \leq$

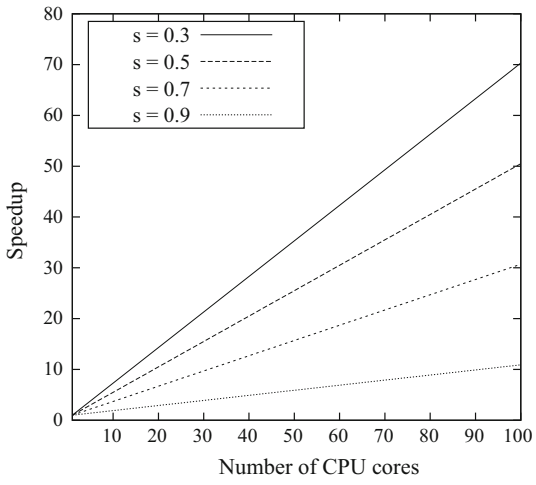

Fig. 1 The symmetric scaled speedup as a function of number of CPU cores for various values of s

$k_c \leq 1$). In the case of a symmetric processor, one core is active during the sequential computation and consumes a power of 1, while the remaining $(c - 1)$ CPU cores consume $(c-1)k_c$. During the sequential computation period, the processor consumes a power of $1+(c-1)k_c$. Thus, during the parallel computation time period, c CPU cores consume c power. It requires s and $(1-s)$ to execute the sequential and parallel codes, respectively, so the formula for the average power consumption W_s of a symmetric processor is as follows.

$$W_s = \frac{s \cdot \{1 + (c-1) \cdot k_c\} + (1-s) \cdot c}{s + (1-s)} \tag{2}$$

Next, we define the *scaled performance per watt (Perf/W)* metric to represent the amount of performance that can be obtained from 1 watt of power. *Perf/W* is basically the reciprocal of energy. The *Perf/W* of a single CPU core execution is 1, so the scaled $Perf/W_s$ achievable for a symmetric processor is formulated as follows.

$$\frac{Perf}{W_s} = \frac{Speedup_s}{W_s} = \frac{c + (1-c) \cdot s}{s \cdot \{1 + (c-1) \cdot k_c\} + (1-s) \cdot c} \tag{3}$$

Figure 2 plots the scaled performance per watt for a symmetric multi-core processor as modeled by Eq. (3), showing that the performance per watt decreases rapidly for a small number of cores. However, as the number of cores increases, so does the problem size, and the inherently serial portion becomes much smaller as a proportion of the overall problem. Therefore, the performance per watt remains almost constant as the number of cores increases and reflects the assumption that the execution time remains fixed.

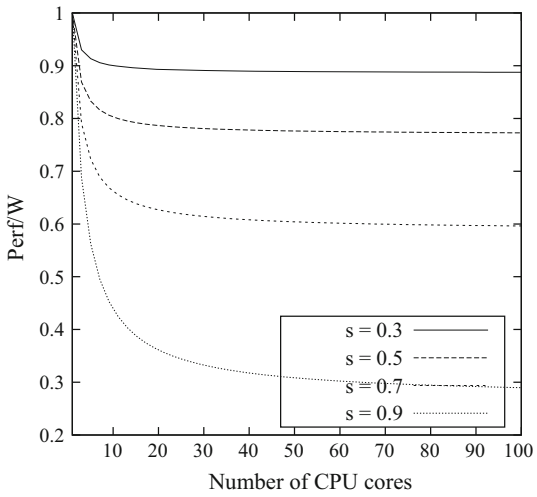

Fig. 2 Scaled performance per watt as a function of the number of CPU cores of a symmetric multi-core processor when $k_c = 0.3$

3.3 Symmetric Scaled Performance Per Joule

The definition of scaled *Perf* / *W* metric allows us to evaluate the performance achievable by a derived unit of power (watt). Power is the rate at which energy is converted, so we can define a *Performance per Joule (Perf/J)* metric where the joule is the derived unit of energy, representing the amount of performance stored in an electrical battery. The *Perf* / *J* of a single CPU core execution is 1, so the scaled *Perf* / J_s achievable by a symmetric processor is formulated as follows.

$$\begin{aligned} \frac{Perf}{J_s} &= Speedup_s \cdot \frac{Perf}{W_s} \\ &= \frac{\{c + (1-c) \cdot s\}^2}{s \cdot \{1 + (c-1) \cdot k_c\} + (1-s) \cdot c} \end{aligned} \tag{4}$$

Figure 3 plots the scaled performance per joule for a symmetric multi-core processor as a function of the number of CPU cores. It can be observed that parallelism costs a substantial amount of energy, which increases linearly as the number of cores increases. The increase in performance matches the increase in power consumption, so the scaled performance per joule increases linearly. For simplicity, hereafter we will omit the prefix *scaled* from the metric definitions.

4 Asymmetric CPU–GPU Processors

In this section, an asymmetric CPU–GPU processor **where CPU and GPU cores are integrated on the same die and share the same memory space and power budget** will be referred to as a *hybrid processor*.

We assume that a program's execution time can be composed of a time period where the program runs sequentially (s), a time period where the program runs in parallel

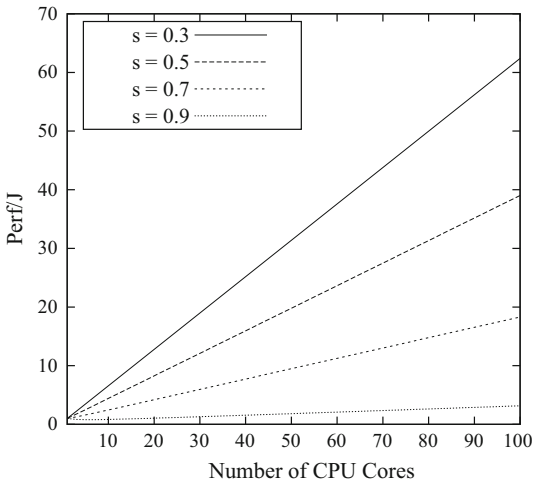

Fig. 3 Scaled performance per joule as a function of the number of CPU cores in a symmetric multi-core processor when $k_c = 0.3$

on the CPU cores (α), and a time period where the program runs in parallel on the GPU cores ($1 - \alpha$). Note that in this case it is assumed that **the program runs in parallel on the CPU cores *or* on the GPU cores, but not on both at the same time**. Simultaneous asymmetric processing will be the topic of the next section.

To model the power consumption of an asymmetric processor we introduce another variable, k_g, to represent the fraction of power a single GPU core consumes in its idle state ($0 \leq k_g \leq 1$). We introduce two further variables, α and β, to model the performance difference between a CPU core and a GPU core. The first variable represents the fraction of a program's execution time that is parallelized on the CPU cores ($0 \leq \alpha \leq 1$), while the second variable represents a GPU core's performance normalized to that of a CPU core ($0 \leq \beta$). For example, comparing the performance of a single CPU core (Intel Core-i7-960 multi-core processor) against the performance of a single GPU core (NVIDIA GTX 280 GPU processor) yields values of β between 0.4 and 1.2. Furthermore, recent studies such as [27] shows that a GPU processor (NVIDIA GTX 280) achieves only 2.5× speedup in average compared to a multi-core processor (Intel Core-i7-960).

We assume that one CPU core in an active state consumes a power of 1 and the *power budget (PB)* of a processor is 100. Thus, $g = (PB - c)/w_g$ is the number of the GPU cores embedded in the processor where variable w_g represents the active GPU core's power consumption relative to that of an active CPU core ($0 \leq w_g$).

4.1 Asymmetric Speedup

Now, if the sequential code of the program is executed on a single CPU core the following equation represents the theoretical achievable *asymmetric speedup* ($speedup_a$).

$$Speedup_a = s + N \cdot (1 - s) \cdot \left\{ \alpha \cdot c + \frac{(1 - \alpha) \cdot g}{\beta} \right\} \tag{5}$$

where N is the number of hybrid processors. Each hybrid processor contains c CPU cores and g GPU cores.

Figure 4 shows the speedup of an asymmetric processor as a function of the number of hybrid processors and as a function of CPU cores within each hybrid processor.

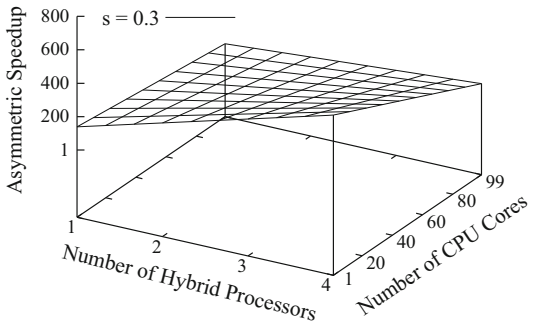

Fig. 4 Asymmetric speedup as a function of the number of hybrid processors and various CPU–GPU chip configurations for $s = 0.3$, $w_g = 0.25$, $\alpha = 0.5$ and $\beta = 1.0$

When an abundance of parallelism is available ($s = 0.3$), the speedup increases linearly with the increase in the number of hybrid processors. Moreover, the maximum speedup is obtained for a chip configuration of 4 hybrid processors, 1 CPU core and 396 GPU cores.

4.2 Asymmetric Performance Per Watt

To model the power consumption of an asymmetric processor we assume that during the sequential computation phase, one CPU core is in active state and the amount of power it consumes is 1, the $c - 1$ idle CPU cores consume $(c - 1)k_c$ and the g idle GPU cores consume $g \cdot w_g \cdot k_g$. During the parallel computation on the CPU cores, the CPU cores consume c and the g idle GPU cores consume $g \cdot w_g \cdot k_g$. During the parallel computation on the GPU cores, the GPU cores consume $g \cdot w_g$ and the idle CPU cores consume $c \cdot k_c$.

Let P_s, P_c, and P_g denote the power consumption during the sequential, CPU, and GPU processing phases, respectively.

$$
\begin{aligned}
P_s &= s \cdot \{1 + (c - 1) \cdot k_c + g \cdot w_g \cdot k_g\} \\
P_c &= \alpha \cdot (1 - s) \cdot \{c + g \cdot w_g \cdot k_g\} \\
P_g &= (1 - \alpha) \cdot (1 - s) \cdot \{g \cdot w_g + c \cdot k_c\}
\end{aligned}
$$

It requires time $(1 - s)$ to perform the parallel computation, which is the sum of times $\alpha \cdot (1 - s)$ and $(1 - \alpha) \cdot (1 - s)$ to perform the parallel computations on the CPU and GPU, respectively, so the average power consumption W_a of an asymmetric processor is as follows.

$$
W_a = P_s + P_c + P_g \tag{6}
$$

Consequently, $Perf/W_a$ of N asymmetric processors is expressed as

$$
\frac{Perf}{W_a} = \frac{s + N \cdot (1 - s) \cdot \left\{\alpha \cdot c + \frac{(1-\alpha) \cdot g}{\beta}\right\}}{P_s + N \cdot (P_c + P_g)} \tag{7}
$$

Figure 5 shows the performance per watt of an asymmetric processor for s = 0.3 as a function of the number of hybrid processors and as a function of CPU cores within each hybrid processor. It can be seen that the $Perf/W_a$ decreases slowly with the increase in the number of hybrid processors, as expected, and decreases faster as the number of the CPU cores increases. Furthermore, the optimal $Perf/W_a$ is obtained for a chip configuration of 4 hybrid processors, 1 CPU core and 396 GPU cores.

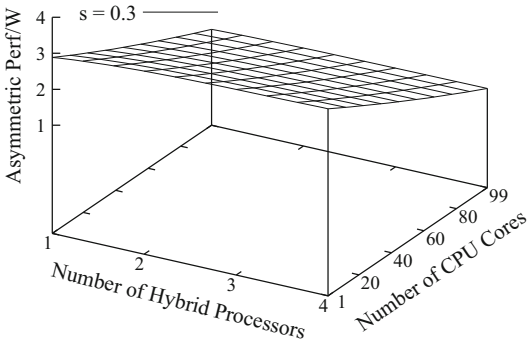

Fig. 5 Asymmetric perf/W as a function of the number of hybrid processors and various CPU–GPU chip configurations for $s = 0.3$, $w_g = 0.25$, $\alpha = 0.5$, $k_c = 0.3$, $k_g = 0.2$ and $\beta = 1.0$

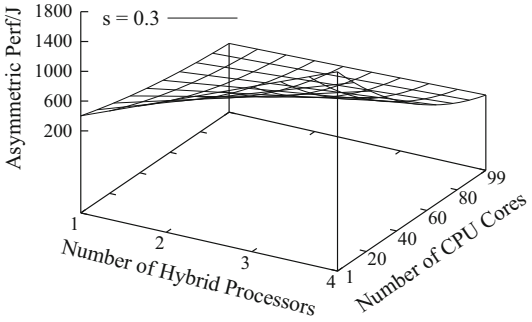

Fig. 6 Asymmetric perf/J as a function of the number of hybrid processors and various CPU–GPU chip configurations for $s = 0.3$, $w_g = 0.25$, $\alpha = 0.5$, $k_c = 0.3$, $k_g = 0.2$ and $\beta = 1.0$

4.3 Asymmetric Performance Per Joule

Based on our definition of performance per joule, the $Perf/J_a$ of N asymmetric processors is expressed as follows.

$$\frac{Perf}{J_a} = \frac{\left\{s + N \cdot (1-s) \cdot \left(\alpha \cdot c + \frac{(1-\alpha)\cdot g}{\beta}\right)\right\}^2}{P_s + N \cdot (P_c + P_g)} \tag{8}$$

Figure 6 shows the performance per joule of an asymmetric processor for s = 0.3 as a function of the number of hybrid processors and as a function of CPU cores within each hybrid processor. It can be observed that the $Perf/J_a$ increases dramatically with the increase in the number of hybrid processors and reaches peak performance for a chip configuration of a single CPU core. On the other hand, the $Perf/J_a$ decreases extremely fast once the GPU cores become dominant.

5 CPU–GPU Simultaneous Processing

In the previous analysis we assumed that a program's execution time is divided into three phases as follows: a *sequential phase* where one core is active, a *CPU phase* where the parallelized code is executed by the CPU cores, and a *GPU phase* where

the parallelized code is executed by the GPU cores. However, the aim of hybrid CPU–GPU computing is to divide the program while allowing the CPU and the GPU will execute their codes simultaneously.

5.1 Simultaneous Asymmetric Speedup

We conduct our analysis assuming that the CPU's execution time overlaps with the GPU's execution time. Such an overlap occurs when the CPU's execution time $\alpha \cdot p \cdot c$ equals the GPU's execution time $\frac{(1-\alpha)\cdot p \cdot g}{\beta}$. Let α' denote the value of α that applies to this equality:

$$\alpha' = \frac{g}{g + c \cdot \beta}$$

We assume that the sequential code of the program is executed on a single CPU core. Thus, the following equation represents the theoretical achievable *simultaneous asymmetric speedup* ($speedup_{sa}$):

$$\begin{aligned} Speedup_{sa} &= s + N \cdot (1-s) \cdot \{\alpha' \cdot c\} \\ &= s + N \cdot (1-s) \cdot \left\{ \frac{(1-\alpha') \cdot g}{\beta} \right\} \end{aligned} \tag{9}$$

where N is the number of hybrid processors. Each hybrid processor contains c CPU cores and g GPU cores.

Figure 7 shows the simultaneous speedup of an asymmetric processor for $s = 0.3$ as a function of the number of hybrid processors and as a function of CPU cores within each hybrid processor. It can be observed that the $Speedup_{sa}$ increases slowly with the increase in the number of hybrid processors. The speedup curve increases very sharply for chip configurations combining a small number of CPU cores with many GPU cores. The superior scalability of the GPU cores is the source of the high performance computation obtained. As the value of α decreases, the time period parallelized by the GPU increases and speedup is also increased. The speedup then begins to increase slowly as the number of CPU cores is increased (and the number of GPU cores is

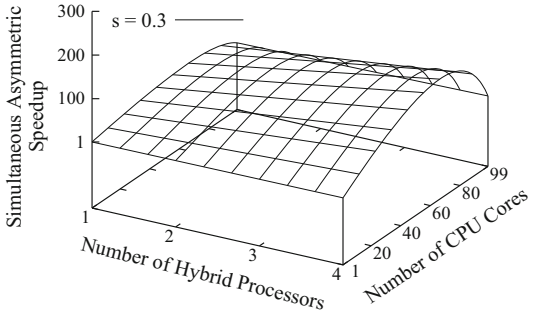

Fig. 7 Simultaneous asymmetric speedup as a function of the number of hybrid processors and various CPU–GPU chip configurations for $s = 0.3$, $w_g = 0.25$ and $\beta = 1.0$

decreased) until it reaches the point beyond which the speedup decreases fast where the dominance of the GPU cores is negligible. For example, for $\alpha = 0.5$, $\beta = 1.0$ and $s = 0.3$ (Fig. 7) the peak performance occurs at the configuration point where CPU cores = 80, GPU cores = 80 and hybrid processors = 4.

5.2 Simultaneous Asymmetric Perf/W

To model the power consumption of an asymmetric processor in a simultaneous processing mode, we assume that one core is active during the sequential computation and consumes a power of 1, while the remaining $c - 1$ idle CPU cores consume $(c - 1)k_c$ and g idle GPU cores consume $g \cdot w_g \cdot k_g$. Thus, during the parallel computation time period, c active CPU cores consume c and g active GPU cores consume $g \cdot w_g$. It requires $(1 - p)$ to execute sequential code and $\alpha' \cdot p$ to execute the parallel codes on the CPU and the GPU simultaneously, so the average power consumption of an asymmetric processor in a simultaneous processing mode is

$$W_{sa} = P_s + P_c + P_g \tag{10}$$

where

$$P_s = s \cdot \{1 + (c - 1) \cdot k_c + g \cdot w_g \cdot k_g\}$$
$$P_c + P_g = \alpha' \cdot (1 - s) \cdot \{c + g \cdot w_g\}$$

Consequently, $Perf/W_{sa}$ of N asymmetric processors in a simultaneous processing mode is expressed as follows.

$$\frac{Perf}{W_{sa}} = \frac{s + N \cdot (1 - s) \cdot \{\alpha' \cdot c\}}{P_s + N \cdot (P_c + P_g)} \tag{11}$$

Figure 8 shows the performance per watt of an asymmetric processor, as modeled by Eq. (11), for $s = 0.3$ as a function of the number of hybrid processors and as a function of CPU cores within each hybrid processor. It can be observed that the $Perf/W_{sa}$ slightly decreases with the increase in the number of hybrid processors.

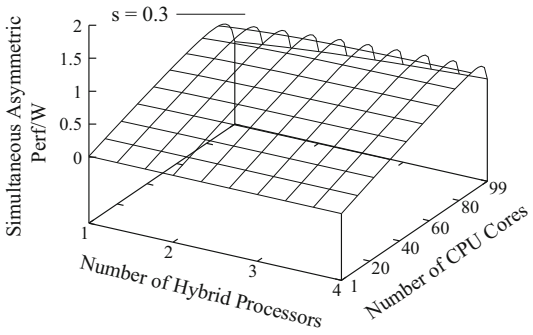

Fig. 8 Simultaneous asymmetric Perf/W as a function of the number of hybrid processors and various CPU–GPU chip configurations for $s = 0.3$, $w_g = 0.25$, $k_c = 0.3$, $k_g = 0.2$ and $\beta = 1.0$

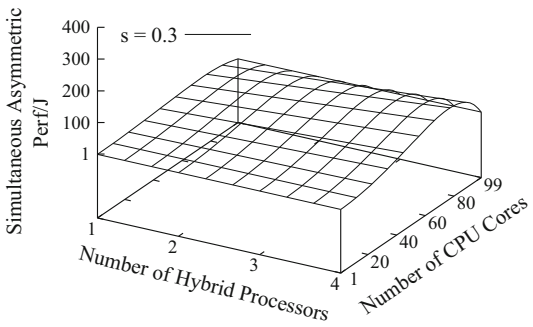

Fig. 9 Simultaneous asymmetric Perf/J as a function of the number of hybrid processors and various CPU–GPU chip configurations for $s = 0.3$, $w_g = 0.25$, $k_c = 0.3$, $k_g = 0.2$ and $\beta = 1.0$

Furthermore, it can be seen that the low performance per watt reflects the behavior of the asymmetric speedup (Fig. 7). When the performance of the CPU cores dominates, the graph increases rapidly as the number of CPU cores increases (and the number of GPU cores is decreases). Then, it reaches the point beyond which the performance per watt decreases very rapidly because the dominance of the GPU cores is negligible.

5.3 Simultaneous Asymmetric Perf/J

Based on our definition of performance per joule, the $Perf/J_{sa}$ of N asymmetric processors in the simultaneous processing mode is expressed as follows.

$$\frac{Perf}{J_{sa}} = \frac{\{s + N \cdot (1 - s) \cdot (\alpha' \cdot c)\}^2}{P_s + N \cdot (P_c + P_g)} \tag{12}$$

Figure 9 shows the performance per joule of an asymmetric processor, when the CPU and the GPU are in simultaneous processing mode, for $s = 0.3$ as a function of the number of hybrid processors and as a function of CPU cores within each hybrid processor. It can be observed that the $Perf/J_{sa}$ slightly decreases with the increase in the number of hybrid processors. Moreover, it increases fast with the increase in the number of CPU cores. It continues to do so until it reaches the point beyond which the performance per joule decreases rapidly because the dominance of the GPU cores is reduced.

6 Synthesis

Figure 10 shows the three speedups investigated in this study, i.e., symmetric speedup(s), asymmetric speedup (a), and simultaneous asymmetric speedup (sa) for a single hybrid processor. The main finding from this comparison is that the same chip configurations choices have different impact on the performance of different processing schemes. For example, a chip configuration of a single CPU core offers the best performance while in asymmetric processing mode. On the other hand, in symmetric processing mode the speedup increases linearly with the increase in the

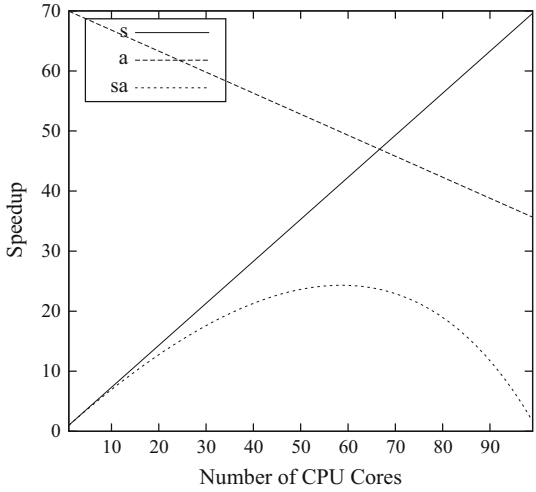

Fig. 10 Symmetric (s), asymmetric (a) and simultaneous asymmetric (sa) speedups as a function of the number of CPU cores for one hybrid processor and for $s = 0.3$, $w_g = 0.25$, $\alpha = 0.5$ and $\beta = 2.0$

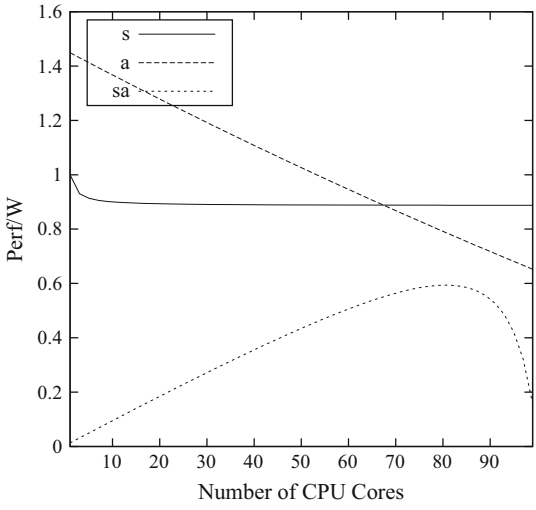

Fig. 11 Symmetric (s), asymmetric (a) and simultaneous asymmetric (sa) Perf/W as a function of the number of CPU cores for one hybrid processor and for $s = 0.3$, $w_g = 0.25$, $\alpha = 0.5$ and $\beta = 2.0$

number of CPU cores while simultaneous processing yields a peak performance at a specific chip configuration of 58 CPU cores and 164 GPU cores. These phenomena indicates that dynamic configuration is required to identify and set the optimal chip organization. Another noticeable observation is that fixed time speedup does not scale when heterogeneous architecture is offered. Therefore, when the problem size has to be increased with the increase in the number of the resources is better to prefer homogeneous architecture.

Figure 11 shows the performance per watt of the three processing schemes that were studied in this research [symmetric (s), asymmetric (a), and simultaneous asymmetric (sa)] and how they are affected by chip configuration. First, it can be observed that the chip configuration has no effect on $Perf/W$ while processing in symmetric mode, as

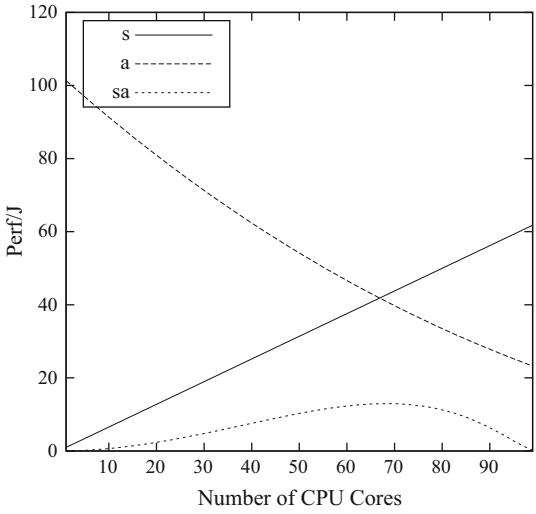

Fig. 12 Symmetric (s), asymmetric (a) and simultaneous asymmetric (sa) Perf/J as a function of the number of CPU cores for one hybrid processor and for $s = 0.3$, $w_g = 0.25$, $\alpha = 0.5$ and $\beta = 2.0$

can be expected. In simultaneous processing mode, $Perf/W$ improves with increasing number of CPU cores until it reaches peak performance for a chip configuration of approximately 85 CPU cores and 60 GPU cores (Fig. 11). Beyond this point, $Perf/W$ decreases rapidly to a point where the contribution of the GPU cores is negligible. On the other hand, in asymmetric processing mode, a chip configuration consisting of a single CPU core yields an optimal performance per watt, and any attempt to increase the number of CPU cores in the chip organization leads to a significant decrease in performance per watt. The performance per watt offered by the simultaneous processing mode is the worst at any configuration setting. Therefore, the best configuration setting for fixed time system in order to achieve the best performance per watt is homogeneous configuration of GPU cores.

Similarly to Figs. 10 and 11, Fig. 12 shows the three performance-per-joule graphs for the analytical models investigated [symmetric (s), asymmetric (a), and simultaneous asymmetric (sa)] and how they are affected by power constraints and chip organization. The first notable finding is the excellent performance-per-joule values that can be obtained by asymmetric processing compared to symmetric and simultaneous asymmetric processing modes. Again, the peak performance occurs for a chip configuration consisting of a single CPU core. The second notable finding is that performance per joule in symmetric and simultaneous asymmetric processing modes reflects the behavior of the speedup and performance per watt of these processing modes. The $Perf/W_s$ has a constant value and therefore is not affected by chip organization, while $Speedup_s$ increases linearly with increasing number of CPU cores. Therefore, $Perf/J_s$ also increases linearly with increasing number of CPU cores. Likewise, $Speedup_{sa}$ and $Perf/W_{sa}$ which exhibit bell curves with maximum values at a specific chip organization, reflect similar behavior for $Perf/J_{sa}$. Moreover, it can be observed again that the performance per joule offered by the simultaneous processing mode is the worst at any configuration setting, and the best performance per joule that a fixed time system can achieved is when it consists of homogeneous GPU cores.

7 Discussions and Future Work

The aim of this section is to summarize our insights from our studies that are presented in [15] and in this paper. In these works we explore the impact of power consumption constrains on current and future heterogeneous many-core processors. We chose to investigate this effect by using *The* two popular *Laws* that are used for performance analysis of parallel systems: Amdahl's law and Gustafson–Barsis's law. The first law is used to model *fixed size* scalable systems where the goal is to solve e a problem as fast as possible when the number of cores increases and *the problem size remains without change*. The second law is used to model *fixed time* large scale systems where the goal is to solve the biggest problem that is possible by increasing the number of cores *while the execution time remains fixed*.

In the case of homogenous architecture, where all the cores are identical, Amdahl's and Gustafson's speedups exhibit increase in performance as the number of cores increases. Gustafson's speedup predicts more optimistic results. These speedups led to steady performance per watt as the system's resources increases and linear increase in the performance per joule. While the curves of the performance per watt of the two models show similarity, the performance per joule modeled by Gustafson's predicts better saving of battery's energy.

As the architecture become heterogeneous while combining fat cores alongside thin cores that share the same die and power budget, the energy efficiency of the architecture is changing dramatically. First, the both laws predict substantial increase in the speedup in case of asymmetric processing mode for configurations of one or a few fat cores and many thin cores. On the other hand, the speedup predicted for simultaneous asymmetric processing mode becomes worst. These results reflect exactly the predicted behavior of the performance per watt and the performance per joule of the two heterogeneous processing modes that were examined. Therefore, we can summarize that heterogeneous system that combined a few CPU cores alongside GPU cores can improves the energy efficiency of the system when operating in asymmetric processing mode but not in simultaneous asymmetric mode.

8 Conclusions

We investigated three analytical models of symmetric, asymmetric, and simultaneous asymmetric processing. These models extended Gustafson–Barsis's Law for symmetric many-core and heterogeneous many-core processors by taking in account various chip organizations and power constraints.

This study of the impact of chip organization on power efficiency and energy consumption has shown that processing in asymmetric mode with a chip configuration consisting of a single CPU core yields outstanding speedup, performance per watt, and performance per joule compared to symmetric and simultaneous asymmetric processing modes. Therefore, asymmetric processing mode offers a substantial improvement in power consumption and energy savings.

The analysis of the three performance metrics with regard to various chip configurations suggest that future many-core processors should be a priori designed to include

one or a few fat cores alongside many efficient thin cores to support energy efficient hardware platforms. Moreover, to achieve optimal scalability and energy savings, a dynamic configuration mechanism is required for identifying and implementing the optimal chip organization.

The work presented in this paper is theoretical. In order to validate the analytical models developed in this work an experimental evaluation is needed. However, the technologies for building heterogeneous processors that allow changing their configuration programmatically do not exist yet. Therefore, the only way to validate our models is by simulations. Fortunately, we have not found an existing simulator that can be served for our purpose. Therefore, we are developing a new simulator these days from the ground up.

References

1. Moore, G.: Cramming more components onto integrated circuits. Electronics **38**(8), 114–117 (1965)
2. Woo, D.H., Lee, H.S.: Extending Amdahl's law for energy-efficient computing in the many-core era. IEEE Comput. **38**(11), 32–38 (2005)
3. Kumar, R., et al.: Heterogeneous chip multiprocessors. IEEE Comput. **38**(11), 32–38 (2005)
4. Mantor, M.: Entering the golden age of heterogeneous computing. C-DAC PEEP2008. http://ati.amd.com/technology/streamcomputing/IUCAA_Pune_PEEP_2008
5. Kogge, P., et al.: Exascale Computing Study: Technology Challenges in Achieving Exascale Systems. DARPA, Washington (2008)
6. Fuller, S.H., Millett, L.I.: Computing performance: game over or next level? IEEE Comput. **44**(1), 31–38 (2011)
7. Borkar, S.: Thousand Core Chips: A Technology Perspective. In: Proceedings of 44th Design Automation Conference (DAC 07), ACM Press, pp. 746–749 (2007)
8. Marowka, A.: Back to thin-core massively parallel processors. IEEE Comput. **44**(12), 49–54 (2011)
9. Krishnamurthy, R.K., Kaul, H.: Ultra-low voltage technologies for energy-efficient special-purpose hardware accelerators. Intel Technol. J. **13**(4), 100–117 (2009)
10. Hillis, D.: The Pattern on the Stone: The Simple Ideas that Make Computers Work. Basic Books, New York (1998)
11. Shi, Y.: Reevaluating Amdahl's law and Gustafson's law. http://www.cis.temple.edu/shi/docs/amdahl/amdahl.html (1996)
12. Amdahl, G.M.: Validity of the Single-Processor Approach to Achieving Large-Scale Computing Capabilities. In: Proceedings of American Federation of Information Processing Societies, AFIPS Press, pp. 483–485 (1967)
13. Gustafson, J.L.: Reevaluating Amdahl's Law. Communications of the ACM, pp. 532–533 (1988)
14. Gustafson, J.L.: The consequences of fixed time performance measurement. Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences, vol. 2, pp. 113–124 (1992)
15. Marowka, A.: Analytical modeling of energy efficiency in heterogeneous processors. Comput. Electr. Eng. J. **39**(8), 2566–2578 (2013)
16. Marowka, A.: Extending Amdahl's law for heterogeneous computing. In: Proceeding of the 2012 10th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA-2012), pp. 309–316
17. Marowka, A.: Modeling the effects of DFS on power consumption in hybrid chip multiprocessors. In: Proceeding of 1st International Workshop on Energy Efficient SuperComputing (E2SC) Held in Conjunction with SC'13, Denver, Colorado, USA, November, 17–22, 2013, ACM digital library
18. Hill, M.D., Marty, M.R.: Amdahl's law in the multicore era. IEEE Comput. **41**(7), 33–38 (2008)
19. Sun, X.H., Chen, Y.: Reevaluating Amdahl's law in the multicore era. J. Parallel Distrib. Comput. **70**, 183–188 (2010)
20. Esmaeilzadeh, H., Blem, E., St. Amant, R., Sankaralingam, K., Burger, D.C.: Dark silicon and the end of multicore scaling. In: Proceeding of 38th International Symposium on Computer Architecture (ISCA), pp. 365–376 (2011)

21. Cho, S., Melhem, R.G.: Corollaries to Amdahl's law for energy. IEEE Comput. Archit. Lett. **7**(1), 25–28 (2008)
22. Cho, S., Melhem, R.G.: On the interplay of parallelization, program performance, and energy consumption. IEEE Trans. Parallel Distrib. Syst. **21**(3), 342–353 (2010)
23. Hong, S., Kim, H.: An integrated GPU power and performance model. In: Proceeding of ISCA10, ACM, pp. 19–23 (2010)
24. Pei, S., Zhang, J., Xiong, N., Kim M.-S., Gaudiot J.-L.: Performance-energy efficiency model of heterogeneous parallel multicore system. In: Green and Sustainable Computing Conference (IGSC), pp. 1–6 (2015)
25. Karanikolaou, E.M., Milovanovic, E.I., Milovanovic, I.Z., Bekakos, M.P.: Performance scalability and energy consumption on distributed and many-core platforms. J. Supercomput. **70**(1), 349–364 (2014)
26. Kim, S.H., Kim, D., Lee, C., Jeong, W.S., Ro, W.W., Gaudiot, J.L.: A performance-energy model to evaluate single thread execution acceleration. Comput. Archit. Lett. **14**(99), 1–4 (2014)
27. Lee, V.W. et al.: Debunking the 100X GPU vs. CPU myth: an evaluation of throughput computing on CPU and GPU. In ISCA'10 Proceedings of the 37th Annual International Symposium on Computer Architecture (2010)